UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/803,514	03/08/2001	Sridhar Obilisetty	026507-000300US	6110

20350 7590 08/24/2007 TOWNSEND AND TOWNSEND AND CREW, LLP TWO EMBARCADERO CENTER EIGHTH FLOOR SAN FRANCISCO, CA 94111-3834	**EXAMINER** VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
08/24/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	09/803,514	OBILISETTY, SRIDHAR
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _19 June 2007_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-36_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-36_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application

6)☐ Other: _____ .

DETAILED ACTION

1.   This action is responsive to the Applicant's response filed 6/19/07.

As indicated in Applicant's response, claims 5, 17 have been amended. Claims 1-36 are

pending in the office action.

Claim Rejections - 35 USC § 103

2.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

3.   Claims 1-12, and 25-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Bloch et al., USPN: 2002/0129129 (hereinafter Bloch) in view of W3C, "SDML – Signed

Document Markup Language", 19, June, 1998 - NOTE-SDML-19980619, pp. 1-36 URL:

http://www.w3.org/TR/1998/NOTE-SDML-19980619/ (hereinafter SDML).

As per claim 1, Bloch discloses a method for implementing an application on a client

computer system, said method comprising code for:

receiving at said client a plurality of text files; each defining a component of the

application (e.g. Fig. 1; steps 66, 68, 70 - Fig. 5; Fig. 6; pg. 10-11, para 0095, 0096);

executing a program resident on said client system for using a combination of said text

files to create an application (e.g. *AVM 221* - Fig. 2; Fig. 7; step 60 Fig. 5); and

creating said application on said client system according to said program (e.g. Fig. 2, 5).

But Bloch does not explicitly disclose checking automatically for updated versions of said text

files. Bloch however addresses the urge for providing latest set of files in accordance to

appropriate version of plug-in files, script file or virtual machine files with regard to version (e.g.

to make sure ... most current versions ... download and install automatically, AVM 221 and its

associated ...plugin -- para 0051-0053, pg. 5-6); hence has taught checking of files and their

latest upgrade; and further discloses parsing XML or HTTP text files and browser (Fig. 5; para

0030, pg. 3). It is noted that a version of a given browser application or instance being conveyed

in the header of Web application files such as XML, HTTP files or any markup type messages

according to W3C standard applicable to data transfer in browser/web technologies was well-

known concept at the time the invention was made (see SDML: sections 1.4, pg. 7; section 4.2,

4.3.1, pg. 13-14; section 4.3.4, 4.3.5, pg. 19-21; DTD pg. 31), in business communication so to

support security aspect of data received. Based on Bloch' s urge to ensure that the latest

associated files to the AVM comes with the application and well-known practices such as W3C

teaching on the multi-platforms usefulness of markup -- like XML/DTD, SDML format-- in Web

messaging/communication, and in light of the desirability of updating browser files to meet the

appropriate virtual machine or execution environment version as mentioned above along with the

compliance checking when markup files are processed by a browser engine as exemplified in

SDML, it would have been obvious for one of ordinary skill in the art at the time the invention

was made to enhance the desire for version checking as shown by Bloch so that using the

browser engine for automatically checking the text files for the latest upgraded file version. One

of ordinary skill in the art would be motivated to do so because this would enable the executing

environment to be provided with the appropriate files according to the version (as set forth by

SDML as an example) expected for such environment as addressed above (see Bloch: para 0051-

0053), such security features being endeavored at length in message across business platforms regarding authenticity of data, and evidenced in part by SDML (see section 1.1, pg. 3-4).

As per claim 2, Bloch discloses XML format (Fig. 1, 2, 4,5).

As per claim 3, Bloch discloses server central source for managing and distributing applications or modifications for applications (e.g. *upgrades, fixes* - pg. 3, para 0032; pg. 5, para 0045-0046; pg. 12, para 0109).

As per claim 4, refer to claim 3 and Bloch's Fig. 1, 2, 4,5.

As per claim 5, Bloch discloses steps of executing an application, sending a request and executing the application in parallel while waiting for response from the request (e.g. ... *reports to the Application Handler 302, ... periodically updates* -- pg. 9, para 0080 – 0082 – Note: resolving a URL with data retrieval while leaving the GUI window on for being updated on tree events changes and notified of download status is equivalent to executing application while waiting for remote response) **without interruption of application functionality** (e.g. *tree contains ... elements 401, corresponding to Files 144 that have been downloaded* - para 0083, pg. 9 - Note: event driven with handlers to update Document tree with dynamic downloaded items reads on functionality of the Document tree application maintained as uninterrupted while the download event occurred)

As per claim 6, Bloch discloses connectionless application execution (e.g. pg. 8, para 0069; pg. 12, para 0108)

As per claim 7, Bloch discloses text files particular to client system (e.g. pg. 4, para 0037; pg. 5, para 0047, 0050)

As per claim 8, Bloch discloses receiving new text file defining a component of said

application, and modifying application by using a newer text files replacing older files

(*upgrades, fixes* - pg. 3, para 0032; *most recent ...version* - pg. 12, para 0107).

As per claim 9, Bloch discloses graphical user interface (e.g. Fig. 6).

As per claim 10, Bloch discloses application being communication preferences for

database invocation (e.g. pg. 7, para 0063; Preference Handler 303 - Fig. 4)

As per claim 11, Bloch discloses data management application (e.g. step *52* – Fig. 5;

Manager 301 -Fig. 4 - Note: downloading files to assemble manager module reads on

application being a management application).

As per claim 12, Bloch discloses component being part of logic of application (pg. 1,

para 0012; pg. 4, para 0037).

As per claim 25, Bloch discloses a computer-readable medium having program code on

a computer system to perform a method comprising code for:

installing a plurality of text files, each defining a component of the application (e.g. Fig.

1; steps 66, 68, 70 - Fig. 5; Fig. 6; pg. 10-11, para 0095, 0096);

installing a program wherein said program comprises instructions for using a

combination of said text files to create an application (e.g. *AVM 221* - Fig. 2); and

creating said application on said client system according to said program (e.g. Fig. 2, 5)

receiving automatically any updated versions of said execution environment files (e.g. *to*

make sure ... most current versions ... download and install automatically -- para 0051-0053,

pg. 5-6) in response to version checking.

But Bloch does not explicitly disclose automatically receiving any updated versions of said text files in response to version checking. Bloch teaches database storage for download support for version upgrades and the implicit version checking by browsers of markup files as set forth in claim 1. Thus, it would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance Bloch deployment and XML processing environment so that not only script or virtual machine files, but also the text files such as the XML files are checked for update and automatic re-download, according to the version checking as known in browser technologies (exemplified by SDML) set forth in claim 1, because of the desirability to conform not only application files but also specification files, a concept inherent to browsers using XML metadata without which format and version conformance would potentially create application execution conflicts; and this has been set forth in the rationale using SDML above in claim 1.

As per claims 26-36, these claims correspond to claims 2-12 respectively, hence are rejected with the corresponding rejections as set forth therein, respectively.

4.   Claims 13-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bloch et al., USPN: 2002/0129129; in further view of Landsman et al., USPN: 6,314,451 (hereinafter Landsman).

As per claim 13, Bloch discloses a computer system with bus, processor coupled to a bus (*Client PC* - Fig. 1; pg. 12, para 0108) for implementing an application comprising the steps:

receiving (text files);

executing (program resident);

creating (application). All these steps limitations have been addressed in the

corresponding office action portions addressing claim 1; hence are rejected herein with the

corresponding rejections as set forth therein, respectively.

But Bloch does not disclose uploading results from using said application to a server

computer system. However, Bloch teaches legacy browsers (para 0048) so as to obviate

redeploying using alternate means as well as remote persisting of records on reuseable user

application data until the user decide to change the application preferences (para 0069-0070, pg.

8). Landsman teaches a browser environment where the user can customize application-related

preferences by providing mouse-clicking interface analogous to the user-driven method of Bloch.

Retransmitting of deployed application results are evidenced further in Landsman's method

wherein logs of application execution data are uploaded to a server (e.g. col. 31, line 62-67).

Based on the desirability to persist user preferences and the implied benefits of legacy of schema

being used for users as mentioned above, it would have been obvious for one of ordinary skill in

the art at the time the invention was made to enhance Bloch server database records so that there

is an uploading of application results as taught by Landsman so that improvement of previous

results or user schema preferences would lend some insight as implied via Bloch's teachings or

via the analysis of logs data by Landsman.

As per claims 14-24, these claims correspond to claims 2-12 respectively, hence are

rejected with the corresponding rejections as set forth therein, respectively.

Response to Arguments

5.   Applicant's arguments filed 6/19/07 have been fully considered but they are not

persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 103 Rejection:

(A)   Applicant has submitted that Bloch's AVM is purported to upgrade just the AVM, not

automatically 'updating' and/or 'upgrading' the text files as claimed (refer to claim 1 for

example), at least not the text files on which the AVM operates (Appl. Rmrks pg. 10, top).

There is no sufficient teaching in the claim language with respect to *version checking* limitation

to enforce that the checked text files is for the AVM to operate on them. Besides, any file being

downloaded and dynamically depicted in the Document tree by Bloch entails downloading of

text files necessary for the AVM to operate. The 103 rationale has been based on the version

checking of a file like a AVM related files (or any downloaded text schema) for this application

building process --provided via the GUI tool by Bloch, in which XML parsing enables creation

of the APP elements for the Document tree based on said schema (see para 0083, pg. 9; Fig. 5, 7

). From there, the rationale utilizes well-known concepts about XML version for a browser, in

conjunction with the urge by Bloch to check proper file version pertinent to a AVM instance (see

para 0051, pg. 5-6; Fig. 1) to put forth the combined teaching including the motivation to do so.

The rejection is deemed satisfactory in terms of *prima facie* establishing of teachings and related

rationale in order to render obvious a claimed limitation. Such 103 rationale as set forth in the

rejection is not intended to discuss whether any of the references anticipate the so-called 'text

files', anticipation which appears to be contested in the argument. First, the Applicant does not

mention about any specific claim (what claim seems to be at stakes here?) when discussing the

'automatically updating/upgrading' concept (such language does not exist in any independent

claim); second, the argument seems to be focusing of a *text file* limitation (Note: XML schema

does read on a text file, for the sake of argument) when the 103 rationale is addressing

'automatically checking' as an obvious limitation; third, the Applicant forgets to mention how as

combined the references still fail to meet this 'version checking' limitation, notwithstanding the

insignificant merits given to the term 'automatically' (see MPEP § 2144.04 (III) when there is

nothing in the claim that describes how *automatically* this *checking* is implemented. The

argument is therefore not persuasive, regardless of what claim, with regard of the version of text

files features.

(B)   Applicant has submitted that the AVM upgrading of para 0053 does not disclose

upgrading of text files (Appl. Rmrks pg. 10, middle). The claims in whole does not explicitly

recite 'upgrading' of text files; and if they did, the teaching by Bloch and the rationale as set

forth in the rejection of claim 1 have been sufficient to provide a combination of teaching leading

to the 103 rejection. The Applicant fails to properly put forth how the above rationale would

have been inapposite, taking under consideration every factual data and suggested subject matter

as set forth by the Office Action when laying out this § 103 rationale (refer to the latter part of

section A).

(C)   Applicant has submitted that Bloch's scheme to occasionally download files when a site

is visited eliminates the need to automatically check for updates and that updating of files by

SDML is *implicit* teaching as inferred by the Office Action (Appl. Rmrks pg. 10, bottom to pg.

11, top). Again, there is no explicit mention of any particular claim or any particular portion of

any claim in the argument. Besides, if a teaching were implicit teaching, the Office Action

would indicate it accordingly. The version checking (of *text files* or any *files* corresponding to

the instance of the development tool by Bloch) has been addressed in claim 1 as an obvious

limitation; and there would be no 103 obvious type of rejection if Bloch has been identified as

already teaching (*implicitly* - as in anticipating via inherency) this upgrade via 'version' of 'text

file' checking. Whether a need to check version is explicit from the claim (recited) scenario (say

of claim 1), this remains a obscure issue. Indeed, the claim does not set forth specifics under

which the version checking is a necessary step taken prior to making use of the text files. From

the claims, the definiteness as to whether these text files are checked prior to download, or before

being incorporated in the creation of the Application remains non-evident or unclaimed; e.g.

from the scenario claim 1(step d), from claim 25 (step d – recited as subsequent to *creating* step

c). There is hence no requirement that a version checking has to be within some specific time

slot or timeframe with respect to downloading or integrating. Arguing that Bloch's checking of

version as redundant teaching would be incorrect because the claim does not enforce a explicit

requirement that would render Bloch's checking to be just so; or that would preclude Bloch's

version checking from being a necessary step. First, the Applicant does not mention about any

specific claim (what claim seems to be at stakes here?) when discussing the 'automatically

updating/upgrading' concept (such language does not exist in any independent claim); second,

the argument seems to be focusing of a *text file* being anticipated (Note: XML schema does read

on a text file, for the sake of argument) when the 103 rationale is addressing 'automatically

checking' as an obvious limitation; third, the Applicant forgets to mention how as combined the

references still fail to meet this 'version checking' limitation (re claim 1), notwithstanding the

lack of weight given to the term 'automatically' (see MPEP § 2144.04 (III) when there is nothing

in the claim that describes how *automatically* this *checking* is implemented. The argument is

therefore not persuasive, regardless of what claim, with regard of the version of text files

features. Bloch is not the sole reference used in addressing this version upgrading/checking

limitation. In response to applicant's arguments against the references individually, one cannot

show nonobviousness by attacking references individually where the rejections are based on

combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

(D)   Applicant has submitted that version in header (of XML files) does not suffice for

application to perform automatic checking of updated versions of files (Appl. Rmrks pg. 11, 2nd

para). The 103 rationale has set forth that Bloch does teach checking of files (and their proper

version) associated with a running instance of a AVM, and when XML schema are needed to

support such application in order to create a target, the version checking as proposed by Bloch

can make use of well-known concept is versioning of XML files so that these schema files would

have to be version-checked prior to or during context of their use for such AVM session. The

Applicant contends that this XML version checking does not need to (does not imply that it) be

automatic just because a (SDML type) header always contains a version. This argument (Appl.

Rmrks pg. 11, second half; pg. 12 top) appears to be a non-persuasive allegation for lacking

further evidence demonstrating how deficient the above reasoning would be; and is not

supporting a clear prima facie rebut whereby the combined teachings as set forth in the 103

rejection would be improper. The need to check files for an instance of a application resides in

Bloch, and the fact that XML files contains version information comes from W3c; so that as

combined it would have been obvious to implement Bloch so to use the common practice in W3c

in order to support Bloch.

(E)   Applicant has submitted (for claim 5) that Bloch does not disclose 'asynchronous mode'

in executing application in parallel with waiting for a response (Appl. Rmrks pg. 12, middle).

There is nothing compelling about this limitation as recited in order to preclude the cited parts by

Bloch from not meeting this 'asynchronous' limitation; and the argument appears to be a mere

allegation for patentability in not providing all the rebut grounds compliant with the CFR §

1.111(b) format.

(F)   Applicant has submitted (for claim 6) that Bloch does not disclose 'connectionless mode'

in executing application in parallel with waiting for a response (Appl. Rmrks pg. 12, bottom, pg.

13, top). There is nothing compelling about this limitation as recited in order to preclude the

cited parts by Bloch from not meeting this 'connectionless' limitation and there is always (at the

time the invention was made) a form of asynchronous mode (in packet transmission) in the very

foundation of TCP/IP communication because otherwise all connections in the entire TCP/IP

network layer would be jammed/stalling because of a obstinate need to await one packet

transmission to reach its destination. Further, the argument appears to be a mere allegation for

patentability in not providing all the rebut grounds compliant with the CFR § 1.111(b) format.

(G)   Applicant has submitted that Bloch's method for download is non-analogous to

Landsman's distribution of HTTP coded advertisements (Appl. Rmrks pg 12 – claims 13-24).

The Bloch/Landsman rationale is for addressing a very defined limitation, that of uploading to

server computer; and in arguing against this rejection, Applicant contends with non-analogous

art approach. In response to applicant's argument that Landsman is nonanalogous art, it has been

held that a prior art reference must either be in the field of applicant's endeavor or, if not, then be

reasonably pertinent to the particular **problem** with which the applicant was concerned, in order

to be relied upon as a basis for rejection of the claimed invention. See *In re Oetiker*, 977

F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In this case, the common endeavor is

downloading of common resources to client of a network; and the particular problem being at

stakes is to use a server to propagate requested resources responsive to any reply/request coming

from the network affiliate; such that both Bloch's browser and Landsman's resource users seem

to share a common need to have a server (as a **problem** to solve in a common endeavor) to

distribute resources; or else for maintaining record of the persisted resources or profile (e.g. via

upload), which after all is analogous to the endeavor of the instant application. The argument is

not persuasive.

In all, the claims will stand rejected as set forth in the Office Action.

Conclusion

6.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The

examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan A Vu
Patent Examiner,
Art Unit 2193
August 22, 2007